

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 09-198223

(43)Date of publication of application : 31.07.1997

(51)Int.Cl. G06F 3/14  
G06F 3/14  
G06F 3/033  
G09G 5/14  
G09G 5/36  
G09G 5/36  
G09G 5/38

(21)Application number : 08-023484

(71)Applicant : CANON INC

(22)Date of filing : 18.01.1996

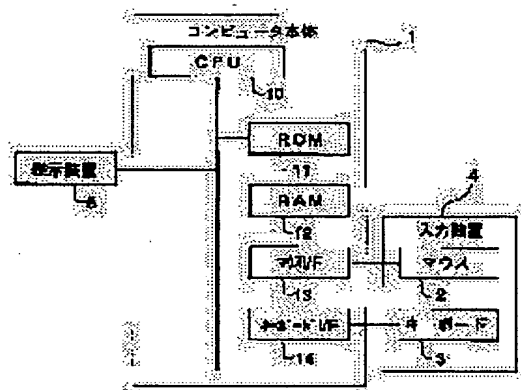
(72)Inventor : ISONUMA TOMOYUKI

## (54) COMPUTER SYSTEM

### (57)Abstract:

**PROBLEM TO BE SOLVED:** To enable an operator to execute comfortable and efficient operation for a graphic object by detecting feature information based upon input information and executing a 1st or 2nd mode based upon the detected feature information.

**SOLUTION:** A CPU 10 executes the 1st mode for changing the state of a graphic object sequentially correspondingly to an input from a mouse 2 or a 2nd mode for changing the state of the graphic object inertially even after the end of the input from the mouse 2 in accordance with a control program stored in a ROM 11. During the execution of the 1st mode, moving speed to be the feature information of the mouse 2 is detected based upon the input information from the mouse 2, and when the detection value exceeds a prescribed value to be a threshold, the 2nd mode is executed. Consequently the operator can execute comfortable and efficient operation for the graphic object.



## LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平9-198223

(43) 公開日 平成9年(1997)7月31日

(51) Int.Cl. <sup>6</sup>	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 3/14	3 4 0		G 0 6 F 3/14	3 4 0 A
	3 5 0			3 5 0 A
	3 8 0		3/033	3 8 0 D
G 0 9 G 5/14		9377-5H	G 0 9 G 5/14	C
5/36	5 2 0	9377-5H	5/36	5 2 0 N

審査請求 未請求 請求項の数 3 F D (全 11 頁) 最終頁に続く

(21) 出願番号 特願平8-23484

(22) 出願日 平成8年(1996)1月18日

(71) 出願人 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(72) 発明者 磯沼 伴幸

東京都大田区下丸子3丁目30番2号 キヤ

ノン株式会社内

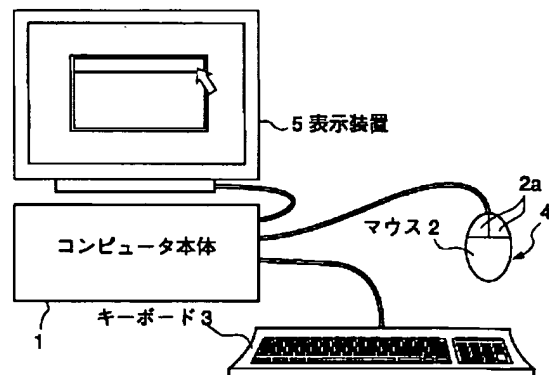
(74) 代理人 弁理士 渡部 敏彦

(54) 【発明の名称】 コンピュータシステム

(57) 【要約】

【課題】 操作者に対し、より快適で効率的なグラフィックオブジェクトの操作を可能とし、1回のドラッグ操作によるオブジェクトの画面上での移動量を従来より拡大したコンピュータシステムを提供する。

【解決手段】 CPUは、マウス2からの入力に逐次対応してグラフィックオブジェクトの状態を変化させる第1のモードを実行中に、マウス2の移動速度が所定値以上になると、マウス2からの入力が終了した後も慣性的にグラフィックオブジェクトの状態を変化させる第2のモードを実行する。



## 【特許請求の範囲】

【請求項1】 操作者との対話にグラフィカルユーザインタフェースを採用し、操作者の移動操作に基づく位置入力装置からの入力情報により、表示画面上でグラフィックオブジェクトの状態を変化させるコンピュータシステムにおいて、

位置入力装置からの入力に逐次対応してグラフィックオブジェクトの状態を変化させる第1のモード、及び位置入力装置からの入力終了した後も慣性的にグラフィックオブジェクトの状態を変化させる第2のモードを実行する実行手段と、

前記入力情報を基にその特徴情報を検出する検出手段と、

前記特徴情報に基づいて前記第1又は第2のモードを実行するよう前記実行手段を制御する制御手段と、  
を有することを特徴とするコンピュータシステム。

【請求項2】 操作者との対話にグラフィカルユーザインタフェースを採用し、操作者の移動操作に基づく位置入力装置からの入力情報により、表示画面上でグラフィックオブジェクトの状態を変化させるコンピュータシステムにおいて、

位置入力装置からの入力に逐次対応してグラフィックオブジェクトの状態を変化させる第1のモード、及び位置入力装置からの入力終了した後も慣性的にグラフィックオブジェクトの状態を変化させる第2のモードを実行する実行手段と、

前記入力情報を基に位置入力装置の移動速度を検出する検出手段と、

前記第1のモードを実行するとともに、この第1のモードの実行中に前記検出手段により検出された移動速度が所定値以上になった場合は前記第2のモードを実行するよう前記実行手段を制御する制御手段と、  
を有することを特徴とするコンピュータシステム。

【請求項3】 前記実行手段は、前記第2のモードを実行する際、各時点のオブジェクトの表示位置を判断し、この判断結果に基づいてグラフィックオブジェクトの慣性運動を停止させることを特徴とする請求項1又は2記載のコンピュータシステム。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】本発明は、操作者との対話にグラフィカルユーザインタフェースを採用したコンピュータシステムに関するものである。

## 【0002】

【従来の技術】コンピュータオペレーティングシステムの利用者インタフェースとして、操作対象となるオブジェクトを視覚化表現して操作するグラフィカルユーザインタフェース（以下「GUI」という。）の環境を持つコンピュータシステムは、既に世の中で一般的な装置として普及しており、その構成は公知である。

【0003】GUI環境においては、マウス等のポインティングデバイスを用いたオブジェクトの操作が高い頻度で発生する。このような操作を行う場合、従来、ドラッグと呼ばれる操作（ドラッグ操作）が用いられてきた。この「ドラッグ操作」とは、ポインティングデバイスに付加されたボタンを押下することにより、操作者が画面上のオブジェクトを選択したことをシステムに伝え、ボタンを押下した状態でポインティングデバイスを移動させることによって、選択されたオブジェクトの状態を連続的に変化させる操作である。

【0004】マウス型のポインティングデバイスでのドラッグ操作の代表例として、図12に示す、ドラッグ操作によるウィンドウの移動操作を挙げることができる。

【0005】また、図形等の描画を行うソフトウェアにおいてもポインティングデバイスを用いたオブジェクトの移動・変形等の操作が頻繁に発生する。この場合も上記ドラッグ操作が用いられる。

## 【0006】

【発明の解決しようとする課題】従来のGUI環境においてグラフィックオブジェクトを画面上で移動させる場合、移動先位置の正確な指定が必要ない操作が存在する。例えば、図12におけるウィンドウの移動操作の目的が、ウィンドウ1（121）とウィンドウ2（122）の重なりを取り除くことにある場合は、ポインティングデバイス（マウス）123を操作して画面124上でカーソル125によるウィンドウ1（121）の移動先位置の正確な指定は必要ではなく、ウィンドウ1（121）の移動先位置はウィンドウ2（122）との重なりが解消する領域であればどこでもよい。

【0007】ところが、従来の方法は、この移動先位置の正確な指定の必要性の有無に拘らず、移動操作が完結するまで操作者がオブジェクトの移動126をポインティングデバイス123の移動127によって制御する必要があった。また、ポインティングデバイス123で最終移動位置を画面124上で指定する必要があるため、その最終移動位置が画面表示領域の外にくるような移動は不可能であった。このため、以下に詳しく説明する2つの問題を生じていた。

【0008】（問題1）移動・回転・縮小拡大等に挙げられるような、その操作がなんらかの数量（移動量、回転角、縮小拡大比）によって特徴づけられるオブジェクトの操作を本発明が問題とする観点から分類すると、以下のようにタイプ1とタイプ2に分類できる。

【0009】タイプ1は、操作の最終目的である状態（移動後の位置、回転後の位置、縮小拡大後の状態）を正確に指定する必要のない操作である。

【0010】タイプ2は、操作の最終目的である状態（移動後の位置、回転後の位置、縮小拡大後の状態）を正確に指定する必要のある操作である。

【0011】問題となるのは、従来のGUI環境におけ

るオブジェクトの操作方法はタイプ2に適した操作方法をタイプ1に対しても同様に適用しており、タイプ1に適した操作方法が存在しなかったことである。

【0012】(問題2)従来のドラッグ操作では、通常、カーソルと呼ばれるポインティングデバイスの物理的位置に対応する点を画面上に表示し、そのカーソルを画面上で移動させながらオブジェクトを移動させていた。このため、図13に示すように、一回のドラッグ操作で移動させることができる最大の移動量は、移動開始時にカーソル132のあった位置から表示画面131の境界133までとなる。つまり、ウィンドウ134は図13の「実際の移動位置」(135)までしか移動できない。

【0013】ところが、ウィンドウが図13の「実際の移動位置」(135)の位置に来たとき、もう一度カーソル133の位置をウィンドウ左角135aに持ち直しドラッグすれば、「原理的に移動可能な位置」(136)までドラッグできる。すなわち、一回のドラッグ操作で移動できる領域は、複数回のドラッグ操作を組み合わせることで移動可能な領域より小さい。この原因はカーソルの移動量とオブジェクトの移動量が等しくなる操作方法を用いているからである。

【0014】そこで、本発明は、上記に鑑みてなされたものであり、操作者に対し、より快適で効率的なグラフィックオブジェクトの操作を可能とし、1回のドラッグ操作によるオブジェクトの画面上での移動量を従来より拡大したコンピュータシステムを提供することを目的とするものである。

【0015】

【課題を解決するための手段】上記目的を達成するために本発明は、操作者との対話にグラフィカルユーザインタフェースを採用し、操作者の移動操作に基づく位置入力装置からの入力情報により、表示画面上でグラフィックオブジェクトの状態を変化させるコンピュータシステムにおいて、位置入力装置からの入力に逐次対応してグラフィックオブジェクトの状態を変化させる第1のモード、及び位置入力装置からの入力終了した後も慣性的にグラフィックオブジェクトの状態を変化させる第2のモードを実行する実行手段と、前記入力情報を基にその特徴情報を検出する検出手段と、前記特徴情報に基づいて前記第1又は第2のモードを実行するよう前記実行手段を制御する制御手段とを有することを特徴とするものである。

【0016】

【発明の実施の形態】以下、本発明の実施の形態を図面を参照して詳細に説明する。

【0017】図1は本発明の実施の一形態に係るコンピュータシステムの外観図である。

【0018】このコンピュータシステムは、コンピュータ本体1を有し、このコンピュータ本体1に、位置入力

装置としてのマウス2及びキーボード3からなる入力装置4、GUI環境でウィンドウ、アイコン等のグラフィカルオブジェクトを表示画面に表示する表示装置5を各々接続している。

【0019】コンピュータ本体1は、図2のブロック図に示すように、本システム全体を制御するCPU10を有し、このCPU10に、CPU10の制御プログラムを記憶するROM11、各種の情報を記憶するRAM12、マウス2が接続されたマウスインタフェース(I/F)13、キーボード3が接続されたキーボードインタフェース(I/F)14、表示装置5を各々接続している。

【0020】マウス2は、二次元型のポインティングデバイスとして現在広く普及しているものである。操作者がマウス2を机上で移動させると、互いに直交する二つの方向に関する移動成分が検知され、この情報がコンピュータ本体1に送信されるようになっている。なお、マウス2が移動量を検知し、これをコンピュータ本体1に送信する構成は既に公知であるため、その説明を省略する。

【0021】また、マウス2は、マウスボタン2aの押下状態と非押下状態を検知する手段と、その変化をマウスI/F13に送信する手段とを有している。これらの手段は既に公知であるため、その説明を省略する。

【0022】なお、本発明は上記で述べたマウス2に限定されるものではなく、ドラッグ操作を行うことのできる全ての位置入力装置(ポインティングデバイス)に適用される。すなわち、本発明の位置入力装置は、位置入力装置の状態として少なくともON状態とOFF状態の2つの状態を持ち、2つの状態を任意に切り替えることができ、さらに、この2つの状態を区別するための情報を、少なくとも状態の変化時にコンピュータ本体1に送る機能を持つ。

【0023】マウスインタフェース(I/F)13は、通常、シリアルインタフェースが適用されるが、これに限定されるものではなく、パラレルインタフェース等の任意の接続方法が可能である。

【0024】ROM11には、図3に示すように、システムソフトウェア110と、オブジェクト運動制御部111を含むアプリケーションソフトウェア112とが記述された図4、図5、図7、図8のフローチャートに示すような制御プログラムが格納されている。

【0025】システムソフトウェア110は、コンピュータの基本的なソフトウェアを指し、通常、システムソフトウェアまたはオペレーティングシステムなどと呼ばれているものであり、概念的には図3に示すように、アプリケーションソフトウェア112とコンピュータ本体(ハードウェア)1の間に位置している。現在、広く普及しているGUI環境を備えたシステムソフトウェアは、入力装置、表示装置、記憶装置等の各種ハードウェア

アの基本的な制御を行う以外、アイコンやウィンドウ等の画像データを表示画面上の任意の位置、任意の領域の描画する機能を持ち、操作者に対し視覚的な操作を提供する。

【0026】オブジェクト運動制御部111は、コンピュータ上で動作するソフトウェアとして実現される。本システムでは、このオブジェクト運動制御部111をアプリケーションソフトウェア112の中のサブルーチンとして実現するが、このオブジェクト運動制御部111をシステムソフトウェア110の中に組み込むことも可能であり、その実現は容易である。

【0027】システムソフトウェア110は、マウス2から送られてくる情報を処理してオブジェクト運動制御部111に渡すようになっている。現在、広く普及しているGUIを実現したシステムソフトウェアは、入力装置からの情報を受信した場合、適切なアプリケーションを選択し、この情報を、選択したアプリケーションに伝達する機能を備えている。この機能は一般に、イベント駆動機構と呼ばれている。本システムは入力装置からの情報の取得を、このシステムソフトウェアのイベント駆動機構を利用して行う。すなわち、本発明では、オブジェクト運動制御部111の機能を含んだアプリケーションソフト112を用意し、このアプリケーションを実行する。システムソフトウェア110は、入力装置4からイベント通知を検知し、そのイベントに関する情報を取得する。そして、これが本アプリケーションソフト112に対して伝達すべき情報であると判断すると、この情報を、システムソフトウェア110とアプリケーションソフト112間の情報伝達のために定められたRAM12上の特定の変数領域に書き込み、本アプリケーションソフト112の動作を記述したプログラムを呼び出す。呼び出されたプログラムの中では、定められたRAM12上の特定の変数領域の情報を読み出すことによって入力装置4からの情報をシステムソフトウェア110を経由して取得することができる。

【0028】CPU10は、ROM11に記憶された制御プログラムに従い、マウス2からの入力に逐次対応してグラフィックオブジェクトの状態を変化させる追従型操作モード(第1のモード)、及びマウス2からの入力終了後も慣性的にグラフィックオブジェクトの状態を変化させる慣性型操作モード(第2のモード)を実行するものであり、追従型操作モードの実行中に、マウス2からの入力情報を基にマウス2の特徴情報としての移動速度を検出し、その検出値が所定値としての閾値V<sub>threshold</sub>以上になった場合は慣性型操作モードを実行するようになっている。

【0029】次に、本システムによるオブジェクトの移動動作を図4のフローチャートに従って説明する。

【0030】まず、アプリケーションソフト112を起動すると、非操作状態に入り、アプリケーションソフト

112は、システムソフトウェア110からのイベント通知を待つ(S1)。なお、「非操作状態」とは、操作者によるマウス操作が発生せず、オブジェクトが停止している状態をいう。

【0031】マウスボタン2aが押下された場合は(S2)、システムソフトウェア110が、アプリケーションソフト112にマウスイベントを通知し、ステップS1の状態から抜け出る。そして、マウスカーソルの位置が移動操作の対象となるオブジェクトの上であれば(S3)、追従型操作モードに入る(S4)。

【0032】この追従型操作モードの実行内容を図5のフローチャートに従って説明する。追従型操作モードにおけるオブジェクト制御方法は通常のマウス2によるドラッグ操作と同じである。ただし、本発明の特徴は追従型操作モードの実行中に常にマウスカーソルの運動状態を記録することである。なお、本発明はシステムソフトウェア110がアプリケーションソフト112に対し、マウス2に関するイベントを通知する機能を持つことと、アプリケーションソフト112の実行がシステムソフトウェア110からのイベント通知によって再開されることを仮定するが、この仮定は現在のほとんどのGUIを実現するシステムソフトウェア110に備えられている。また、本発明は、アプリケーションソフト112中にマウス2に関するイベントを監視するループを組み込むことにより、システムソフトウェア110にイベント管理機構が備わっていない場合でも適用可能となる。

【0033】まず、追従型操作モードに移行すると、図4と同様、アプリケーションソフト112は、システムソフトウェア110からのマウスイベントの通知を待つ(S41)。

【0034】マウス移動(マウスボタンON)のイベント通知を受け取った場合に(S42)、マウス2が移動していれば(S43)、マウス2の移動先の位置をシステムソフトウェア110から取得し(S44)、直前のマウス2の位置からの移動量を求める。そして、移動量から移動速度を計算する。この移動速度は、後の慣性型操作モードの選択判断で、速度絶対値と閾値の比較に使用するため、正確に、図5及び以下に示す式(1)で求めてもよいが、速度の二乗による次の式(2)、又は空間的な等方性は失われるが、次の式(3)により求めてもよい。

【0035】

【数1】

$$V = \sqrt{\Delta X^2 + \Delta Y^2} \quad \cdots \text{式(1)}$$

【0036】

【数2】

$$V^2 = \Delta X^2 + \Delta Y^2 \quad \cdots \text{式(2)}$$

【0037】

【数3】

$$\max(|\Delta X|, |\Delta Y|) \quad \dots \text{式(3)}$$

$|\Delta X|, |\Delta Y|$ の最大値を選択する

また、現在のマウス位置は後の計算のためにRAM12に記憶しておく。

【0038】以上を行うと、通常のドラッグ操作に必要な処理を行う(S45)。

【0039】この処理は、公知であるため図6を参照し、簡単に説明する。まず、新しい位置X、YにオブジェクトA(51)を再描画し、過去に描かれたオブジェクトA(52)の領域と新しく描かれたオブジェクトA(51)の領域とが重ならない領域(破線で囲んだ領域)53に関して、過去のオブジェクトA(52)によって隠されていたオブジェクトB(54)の一部と背景を再描画する。ここでの「位置X、Yにオブジェクトを描く」という表現は、「オブジェクトの特定の代表点が位置X、Yにくるような位置にオブジェクトを描く」という意味である。さらに、本発明に係る上記描画処理以外に、システムソフトウェア110またはアプリケーションソフト112に本発明に係らない必要な処理がある場合、上記一連の処理を行った後、図5のステップS45の位置で処理する。

【0040】一方、図5の前記ステップS42でマウスボタンOFFのイベント通知を受け取った場合は(S42)、追随型ドラッグ操作の終了と判断し、現在のマウス移動速度を最終速度としてRAM12に記憶し(S46)、追随型操作モードを終了する。

【0041】追随型操作モードが終了すると、オブジェクト運動制御部111は、追随型操作モードの実行中にRAM12に記憶された最終速度情報を解析し、慣性型操作モードを行うか否かを判断する(S6)。なお、この慣性型操作モードの選択判断処理は後に詳細に説明する。

【0042】この判断(S6)後、慣性型操作モードの実行の必要があると判断された場合、慣性型操作モードに移行する(S7)。一方、慣性型操作モードの実行の

$$X = X_{\text{previous}} + V_x, Y = Y_{\text{previous}} + V_y \quad \dots \text{式(4)}$$

ここでの「位置X、Yにオブジェクトを描く」という表現は、「オブジェクトの特定の代表点が位置X、Yにくるような位置にオブジェクトを描く」という意味である。この描画の方法は追随型操作モードでの描画方法と同様である。

【0050】最後に終了条件を調べ(S78)、終了条件を満たしていない場合は、次のタイマ割り込み通知を待ち(S73)、終了条件を満たした場合は、システムソフトウェア110に対して行ったタイマ割り込みの登録を解除し(S79)、慣性型操作モードを終了する。

【0051】ここで、前記ステップS78の慣性操作の終了条件の判断を図8のフローチャートに従って説明する。

必要がないと判断された場合、非操作状態に戻り(S1)、移動操作を終了する。

【0043】慣性型操作モードでは、操作者によるマウス操作が終了した後も、慣性的にオブジェクトを移動させ、特定の終了条件を満たした場合、非操作状態に戻り、移動操作モードを終了する。

【0044】次に、前記ステップS7の慣性型操作モードの実行内容を図7のフローチャートに従って詳しく説明する。

【0045】まず、追随型操作モードの実行中に最終速度としてRAM12に記憶された速度( $V_{x\text{last}}, V_{y\text{last}}$ )を、現在の速度( $V_x, V_y$ )とする(S71)。

【0046】次に、オブジェクトの運動を実現するために、システムソフトウェア110に対し、時間管理用のタイマ割り込みの登録を行う(S72)。すなわち、システムソフトウェア110に対し本アプリケーションソフト112が時間間隔 $\Delta T$ のタイマ割り込みを登録することによって、システムソフトウェア110は本アプリケーションソフト112に対し $\Delta T$ の時間経過ごとにタイマ割り込みイベントを通知する。通常、タイマ割り込みの管理機構はシステムソフトウェア110に備わっているが、仮に備わっていなかったとしても、本アプリケーションソフト112にタイマ管理機構を組み込むことによって、本発明は容易に実現できる。

【0047】システムソフトウェア110に対するタイマ割り込みの登録が完了すると、アプリケーションソフト112は、システムソフトウェア110からのイベント通知待ち状態に入る(S73)。

【0048】システムソフトウェア110からのタイマ割り込み通知が来ると、まず、現在のオブジェクトの位置を以下に示す式(4)によって求め(S74)、その位置にオブジェクトを再描画する(S75)。

【0049】

【数4】

【0052】オブジェクトの次回移動位置を計算する(S781)。オブジェクトが移動できる領域内、例えば、表示画面上の全領域や特定のウィンドウ内において、オブジェクトをドラッグ操作するための最小の部分が領域内に存在するか否かを判断する(S782)。ドラッグ操作を可能とするためには、常にオブジェクトのドラッグ可能な最小部分を移動領域内に残す必要がある。

【0053】次に、現在の移動速度は $V_{\text{stop}}$ 以上になったか否かを判断する(S783)。 $V_{\text{stop}}$ 以上になっていなかった場合は、そこで慣性型操作モードを終了する。 $V_{\text{stop}}$ 以上になった場合は、現在のオブジェクトの表示位置が特定領域内であるか否かを判断する(S78

4)。特定領域内であると判断した場合は、ステップS73に戻る。特定領域内でない判断した場合、すなわち、次に行われる慣性移動処理によって、ドラッグ可能な最小部分が領域外に消えると判断した場合は、そこで慣性型操作モードを終了する。

【0054】なお、オブジェクトの慣性運動に摩擦効果を持たせ、次の式(5)により移動速度を徐々に減速してもよい。なお、式(5)において、 $\alpha$ は減速の割合を決める定数である。この減速処理を行う場合は、図7のフロー中、ステップS77の部分に処理を入れればよい。

【0055】

【数5】

$$V_x \leftarrow V_x - \alpha, V_y \leftarrow V_y - \alpha \quad \dots \text{式(5)}$$

次に、図4の前記ステップS6の慣性型操作モードの選択判断処理の詳細を図9乃至図11を参照して説明する。

【0056】まず、通常のドラッグ操作におけるカーソル位置の移動特性について説明する。まず、ドラッグ操作時のカーソル位置の時間変化について考えてみる。ここでは特にカーソル位置のウィンドウ上でのX座標について考察するが、Y座標についても全く同様の議論が成立する。「発明が解決しようとする課題」で述べたように、従来のドラッグ操作ではタイプ1、タイプ2の操作が存在する。

【0057】タイプ2の操作に関し、ドラッグ操作時のカーソル位置のX座標の軌跡を表すと、図9の902のようになる。この図は一定時間ごとにカーソルの存在する位置に円形のイメージを描画して作成したものである。なお、同図の901はタイプ1の操作に関するものである。

【0058】この図9を基にカーソルの移動位置のX成分の時間変化と、移動速度のX成分すなわち

$V_x = dX/dt$  = 一定時間間隔でのXの変化量の時間変化とをグラフにすると、タイプ2の場合、それぞれ図10の1002、図11の1102となる。

【0059】図11の1102から分かるように、速度変化に関しては、移動操作を始める時点と終了させる時点で、ほぼ $V_x = 0$ となる。開始時点で、ほぼ $V_x = 0$ となるのは、移動開始前にはマウス2は静止しているためであり当然であるが、終了時点で、ほぼ $V_x = 0$ となるのは、移動先の位置を確定する必要がある移動操作においては、操作者が移動先の位置を決めた後に、マウス2の移動操作を終了する必要があるためである。つまり、カーソルが目的の位置より離れている段階では、細かい位置制御が不要であるため、ある程度の速度を持たせてマウス2を移動させることができる。一方、目的位置近傍に来ると、操作者はマウスカーソルの移動速度を緩め、より細かい位置制御をする必要があり、最終的な位置に定まった状態ではマウス2の運動は、ほとんど停

止している。

【0060】タイプ1の操作の場合は、タイプ2の操作とは反対に、カーソルが移動し続ける最中に、すなわち、マウス2を移動させている最中にマウスボタン2aを非押下状態にした場合の軌跡及び速度変化は図10の1001、図11の1101のようになる。

【0061】「発明が解決しようとする課題」のタイプ1の操作、すなわち、操作の最終目的である状態(移動後の位置、回転後の位置、縮小拡大後の状態)を正確に指定する必要のない操作は、図10、図11に示す特性を持ちやすい。なぜなら、タイプ1の操作においては、移動先の位置を確定する必要が無いため、目的位置近傍での緩やかな速度低下は発生しないと考えられる。しかし、操作者がタイプ1の操作を実行しようとした時に、必ずしも、図11の1101のようなタイプ1に期待される特性の操作を行っているとは限らず、図11の1102のようなタイプ2の特定を持つ操作を行うことも考えられる。

【0062】いずれにせよ、本発明を実現する上で重要なことは、操作者がタイプ1の操作を行う時には、図11の1101の特性を持つ操作を行うよう要求しても問題はないということである。

【0063】そこで、本発明では、上述のカーソル移動特性を利用し、操作者が通常のドラッグ操作後、慣性型操作への移行を意図したかどうかをオブジェクト運動制御部111が判断する方法として以下の方法を採用する。すなわち、オブジェクト運動制御部111は、マウスボタン2aが押下状態から非押下状態に変化した時点でのマウス移動速度 $V_{determinant}$ を判断し、その値が特定の閾値 $V_{threshold}$ 以下であれば、「操作者が追従型操作方法を意図してドラッグ操作を終了した」と判断し、閾値 $V_{threshold}$ 以上であれば、「操作者が慣性型操作方法を意図してドラッグ操作を終了した」と判断する。この判断を図5のS6に示す。また、操作者に対して、慣性型操作方法を意図する場合は、マウス2の移動中にマウスボタン2aを離すことを規定し、この操作の区別を要求する。

【0064】以上述べた方法は、人間が現実の世界で実際の物体を静かに移動させる動作と、物体を投げる動作の違いに対応する。すなわち、人間が物体を静かに移動させる場合は、手の動きを物体の移動先の位置上で停止させた後、手を物体から放す。逆に、人間が物体を投げる場合は、手が運動を続けている状態で、手を物体から放す。よって、本発明が導入する慣性型操作モードの選択判断は操作者に対してより自然な感覚として受け入れられると考えられる。

【0065】上述したコンピュータシステムによれば、追従型操作モードでは従来通りのドラッグ操作を行い、追従型操作モード終了後に慣性型操作モードが必要と判断された場合には、操作者によるマウス2の移動操作が止

まった後でも、オブジェクトを慣性的に運動させることによって、従来用いられてきたドラッグ操作によるグラフィックオブジェクトの操作上の無駄が解消され、GUI環境上でのグラフィックオブジェクトの操作が効率的になる。また、一回のドラッグ操作によるオブジェクトの画面上での移動量を拡大することができる。また、従来の追従型操作に対応する第1のモードを継承した上で第2のモードを導入するので、操作者に対し従来の操作方法の変更を要求せず感覚上快適なものとなる。

【0066】なお、本発明のオブジェクト操作方法は、上記に限定されず、他の操作方法に適用するのは容易である。例えば、図形の回転操作であれば、図4、図5、図7、図8のフローにおいて、慣性運動に使用する位置変数、速度変数をそれぞれ、回転角度、回転角速度と解釈すればよい。このように本発明はオブジェクトの移動操作に限定されるものではなく、オブジェクトの操作が移動量、回転量、変形量等の数量によって特徴づけられるもの全てに対し適用可能である。この場合、移動量、回転量、変形量等やこれらの微分値が特徴情報となる。

【0067】

【発明の効果】以上詳述した本発明によれば、位置入力装置からの入力情報の特徴情報に基づいて、位置入力装置からの入力に逐次対応してグラフィックオブジェクトの状態を変化させる第1のモード、又は位置入力装置からの入力が終了した後も慣性的にグラフィックオブジェクトの状態を変化させる第2のモードを実行するので、従来用いられてきたドラッグ操作によるグラフィックオブジェクトの操作上の無駄が解消され、GUI環境上でのグラフィックオブジェクトの操作がより快適で効率的になり、また、1回のドラッグ操作によるオブジェクトの画面上での移動量を従来より拡大することができる。

【図面の簡単な説明】

【図1】本発明のコンピュータシステムの外觀図である。

【図2】本発明のコンピュータシステムのブロック図で

ある。

【図3】本発明の制御プログラムの内容を説明するための概念図である

【図4】本発明のオブジェクトの移動動作を示すフローチャートである。

【図5】本発明の追従型操作モードの詳細を示すフローチャートである。

【図6】本発明のオブジェクト再描画処理を説明するための図である。

【図7】本発明の慣性型操作モードの詳細を示すフローチャートである。

【図8】本発明の慣性型操作モードの終了条件を示すフローチャートである。

【図9】ドラッグ操作時のマウスカーソルの軌跡の実測図である。

【図10】ドラッグ操作時のマウスカーソルのX座標の時間変化を示す図である。

【図11】ドラッグ操作時のマウスカーソルの移動速度X座標成分の時間変化を示す図である。

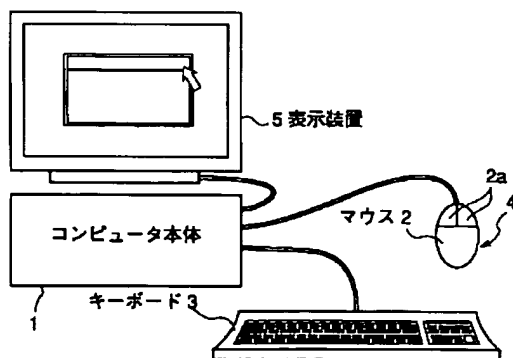
【図12】ウィンドウのドラッグ操作を示す図である。

【図13】従来のドラッグ操作時の移動量の限界を示す図である。

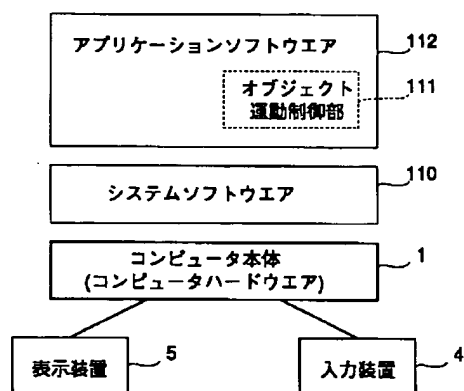
【符号の説明】

- 1 コンピュータ本体
- 2 マウス
- 2a マウスボタン
- 3 キーボード
- 4 入力装置
- 5 表示装置
- 10 CPU
- 11 ROM
- 12 RAM
- 110 システムソフトウェア
- 111 オブジェクト運動制御部
- 112 アプリケーションソフトウェア

【図1】

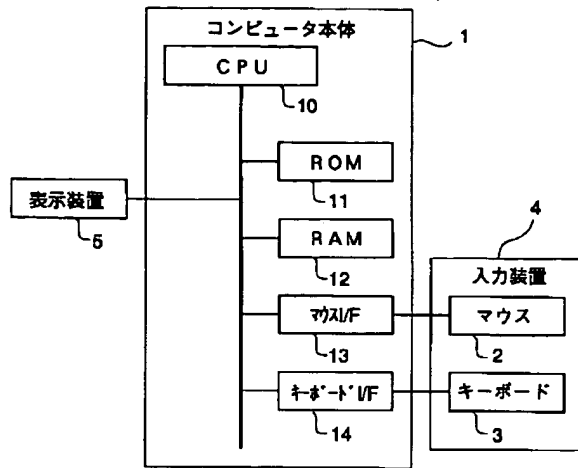


【図3】

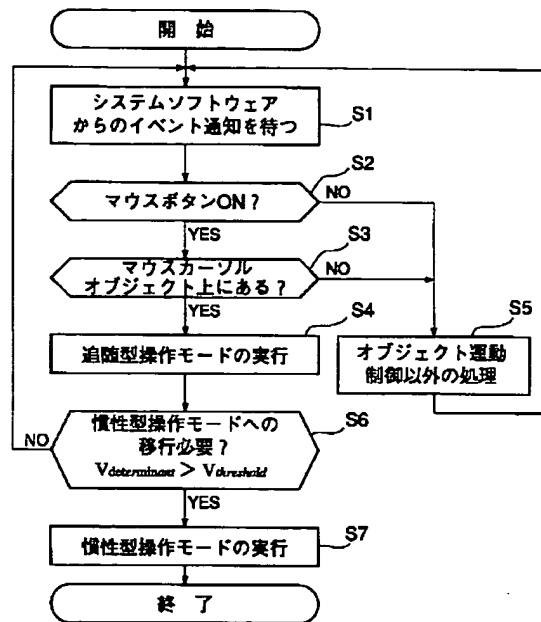




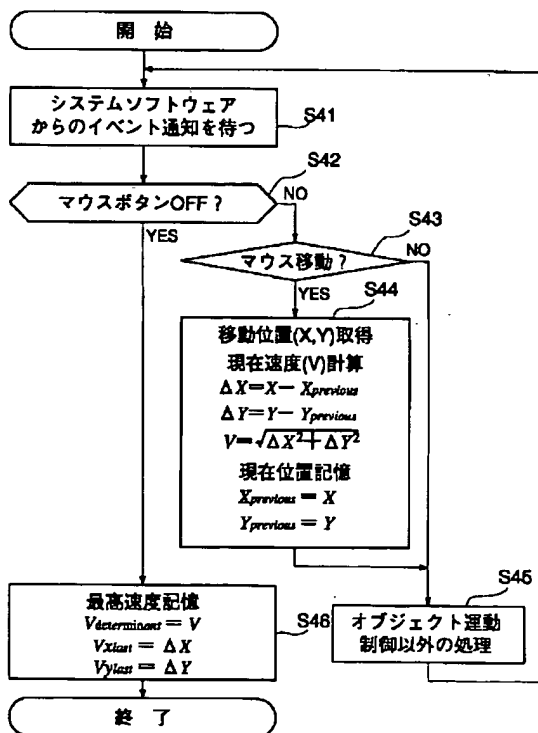
【図2】



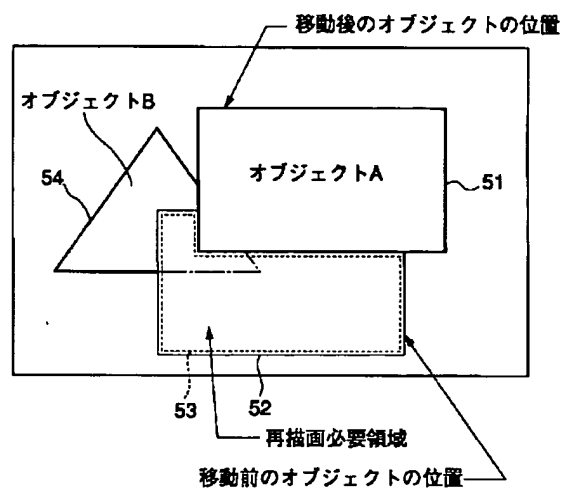
【図4】



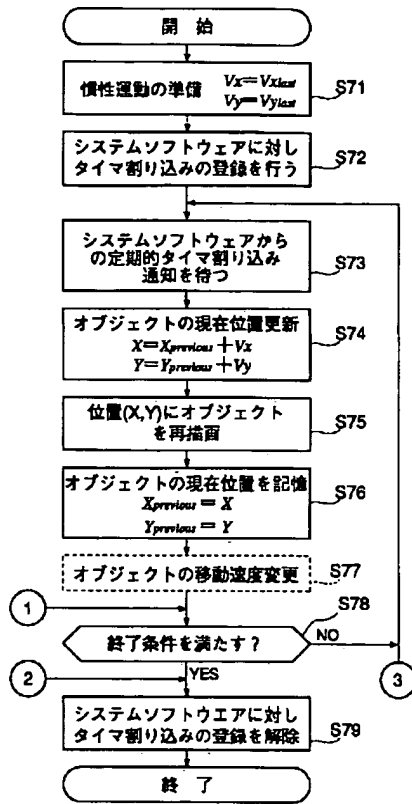
【図5】



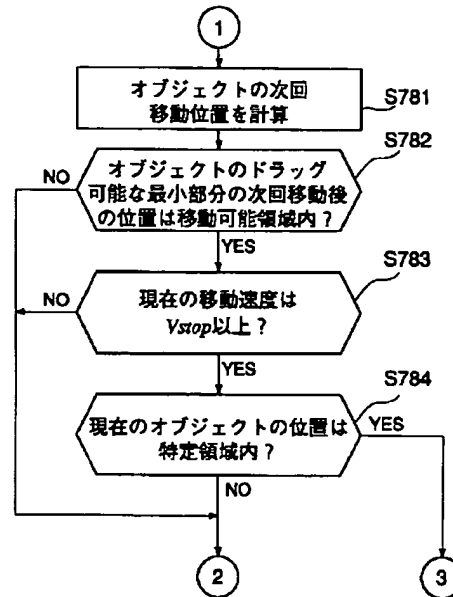
【図6】



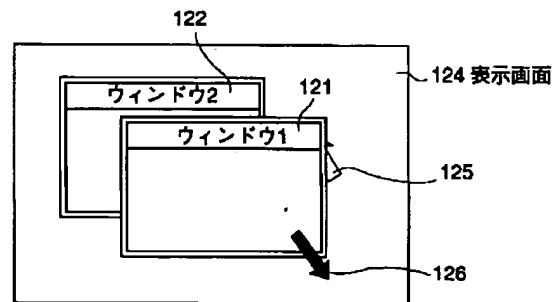
【図7】



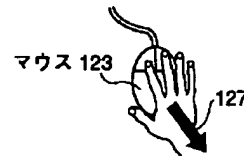
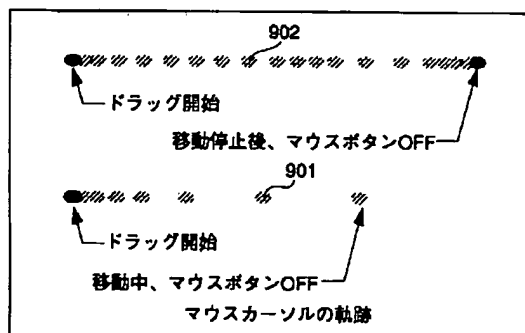
【図8】



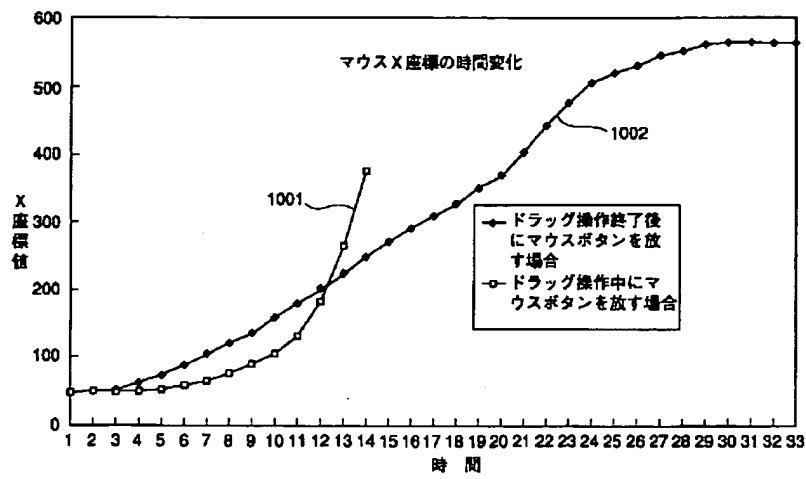
【図12】



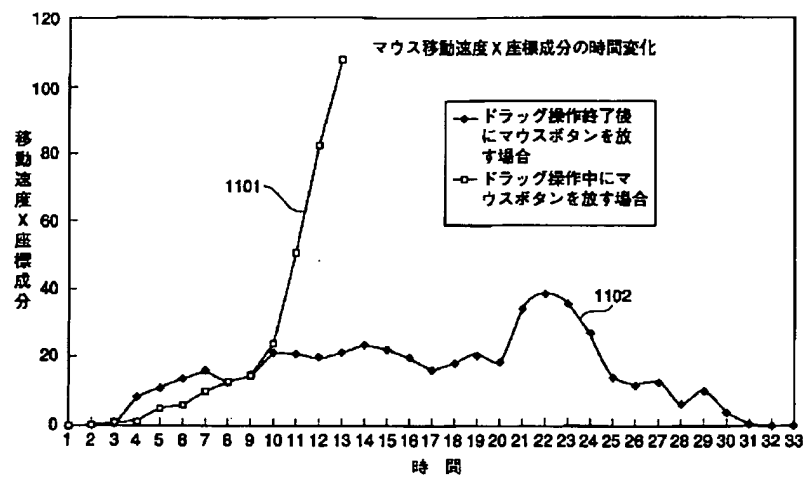
【図9】



【図10】



【図11】



(51)Int.Cl. <sup>6</sup>	識別記号	片内整理番号	F I	技術表示箇所
G 0 9 G 5/36	5 3 0	9377-5H	G 0 9 G 5/36	5 3 0 Y
5/38		9377-5H	5/38	A